**Exercises: Introduction to AJAX**

Problems for exercises and homework for the [“JavaScript Advanced” course @ SoftUni](https://softuni.bg/trainings/2248/js-advanced-february-2019).

## Bus Stop

### Write a JS program that retries arrival times for all buses by given bus stop ID when a button is clicked. Use the

### following HTML template to test your code:

### HTML and JavaScript Code

You are given the following **HTML** and **CSS**.

|  |
| --- |
| buses.html |
| <!DOCTYPE **html**> <**html lang="en"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>Bus Stop</**title**>  <**style**>  **#stopName {**  **font-size: 1.5em;**  **font-weight: 400;**  **padding: 0.25em;**  **background-color: aquamarine;**  **}**  </**style**>  <**script src="https://code.jquery.com/jquery-3.1.0.min.js"  integrity="sha256-cCueBR6CsyA4/9szpPfrX3s49M9vUU5BgtiJj06wt/s="  crossorigin="anonymous"**></**script**> </**head**> <**body**>  <**div** **id**="**stopInfo**" **style**="**width**:**20em**">  <**div**>  <**label** **for**="**stopId**">Stop ID: </**label**>  <**input** **id**="**stopId**" **type**="**text**">  <**input** **id**="**submit**" **type**="**button**" **value**="**Check**" **onclick**="*getInfo*()">  </**div**>  **<div id="result">**  <**div** **id**="**stopName**"></**div**>  <**ul** **id**="**buses**"></**ul**>  </**div**>  </**div**>  <**script**>  **function** *getInfo*() {  // ***TODO*** ...  }  </**script**> </**body**> </**html**> |

When the button with ID **'submit'** is clicked, the name of the bus stop appears and the list bellow gets populated

with all the buses that are expected and their time of arrival. Take the **value** of the input field with id **'stopId'**.

Submit a **GET** request to **https://judgetests.firebaseio.com/businfo/{*stopId*}.json** (replace the

highlighted part with the correct value) and parse the response. You will receive a JSON object in format:

**stopId: {**

**name: stopName,**

**buses: { busId: time, … }**

**}**

Place the name property as text inside the div with ID **'stopName'** and each bus as a list item with text:

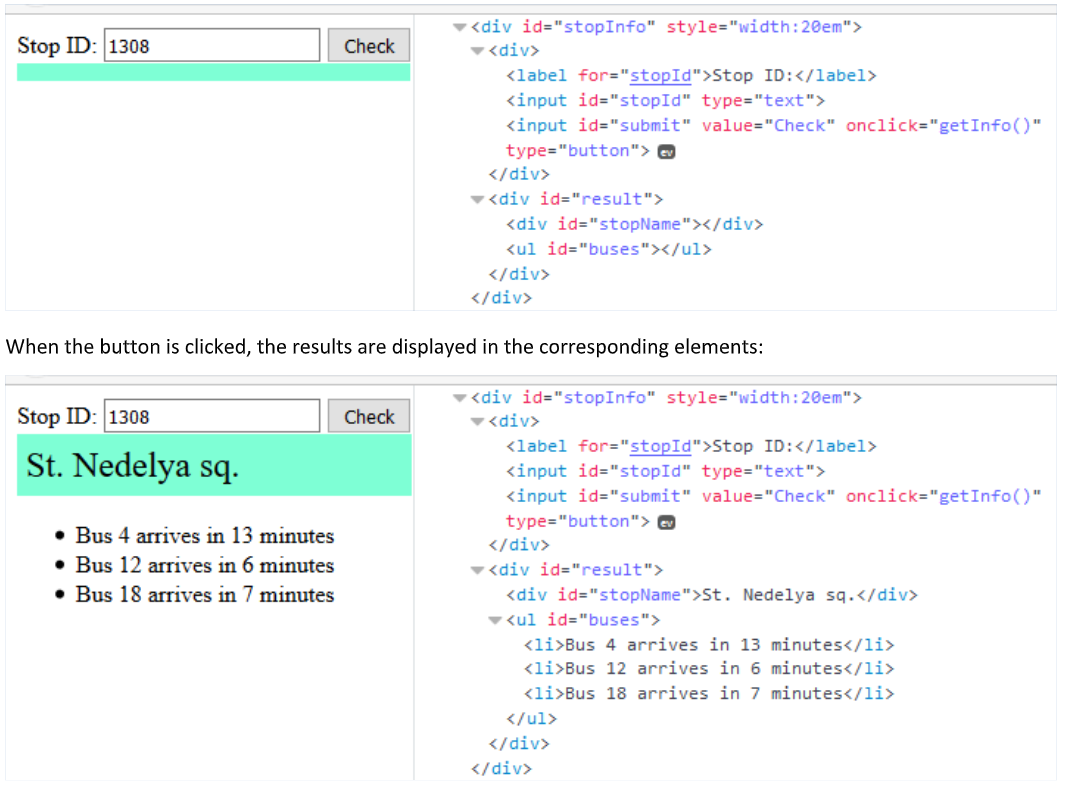
**"Bus {busId} arrives in {time}"**

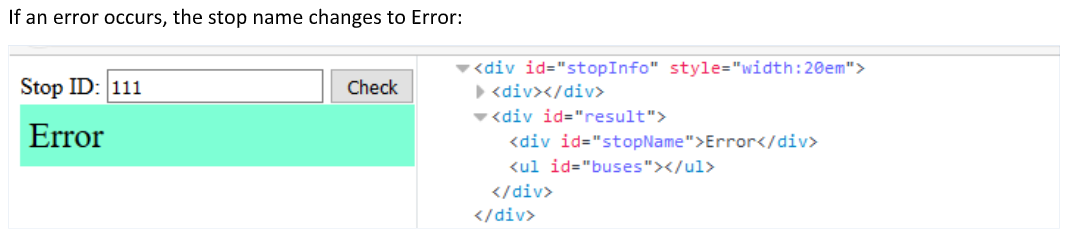
Replace all highlighted parts with the relevant value from the response. If the request is not successful, or the

information is not in the expected format, display **"Error"** as **stopName** and nothing in the list. The list should be

cleared before every request is sent.

Submit only the **getInfo**() function.

**Examples** 



### Hints

The webhost will respond with valid data to IDs 1287, 1308, 1327 and 2334

## Bus Schedule

### Write a JS program that tracks the progress of a bus on it’s route and announces it inside an info box. The program

### should display which is the upcoming stop and once the bus arrives, to request from the server the name of the next

### one. Use the following HTML to test your solution:

### HTML and JavaScript Code

You are given the following **HTML** code:

|  |
| --- |
| schedule.html |
| <!DOCTYPE **html**>  <**html** **lang**="**en**">  <**head**>  <**meta** **charset**="**UTF-8**">  <**title**>Bus Schedule</**title**>  <**style**>  **#schedule** { **text-align**: **center**; **width**: **400px**; }  **input** { **width**: **120px**; }  **#info** { **background-color**:**aquamarine**; **border**:**1px** **solid** **black**; **margin**:**0.25em**; }  **.info** { **font-size**: **1.5em**; **padding**: **0.25em**; }  </**style**>  <**script** **src**="**https://code.jquery.com/jquery-3.1.1.min.js**"></**script**>  </**head**>  <**body**>  <**div** **id**="**schedule**">  <**div** **id**="**info**"><**span** **class**="**info**">Not Connected</**span**></**div**>  <**div** **id**="**controls**">  <**input** **id**="**depart**" **value**="**Depart**" **type**="**button**" **onclick**="***result***.***depart***()">  <**input** **id**="**arrive**" **value**="**Arrive**" **type**="**button**" **onclick**="***result***.***arrive***()"  **disabled**="**true**">  </**div**>  </**div**>  <**script**>  **function** *solve*() {  // *TODO* ...  **return** {  depart,  arrive  };  }  let ***result*** = *solve*();  </**script**>  </**body**>  </**html**> |

The bus has two states – **moving** and **stopped**. When it is **stopped**, only the button “**Depart**” is **enabled**, while the

info box shows the name of the **current** stop. When it is **moving**, only the button “**Arrive**” is **enabled**, while the info

box shows the name of the **upcoming** stop. Initially, the info box shows "**Not** **Connected**" and the "**Arrive**" button is

**disabled**. The ID of the first stop is "**depot**".

When the "**Depart**" button is clicked, make a **GET** request to the server with the ID of the current stop to address

**https://judgetests.firebaseio.com/schedule/{currentId}.json** (replace the highlighted part with

the relevant value). As response, you will receive a JSON object in the following format:

**stopId {**

**name: stopName,**

**next: nextStopId**

**}**

Update the info box with information from the response, disable the “Depart” button and enable the “Arrive”

button. The info box text should look like this (replace the highlighted part with the relevant value):

**Next stop {stopName}**

When the "**Arrive**" button is clicked, update the text, disable the “Arrive” button and enable the “Depart” button.

The info box text should look like this (replace the highlighted part with the relevant value):

**Arriving at {stopName}**

Clicking the buttons in succession will cycle through the entire schedule. If invalid data is received, show "**Error**"

inside the info box and **disable** both buttons.

Submit only the **solve**() function that returns an object, containing the two click event handlers for **depart**() and

**arrive**(), as shown in the sample HTML.

### Examples

Initially, the info box show Not Connected and the arrive button is disabled.

